The SGIA and the Common Growing Language

Laurenţiu CIOVICĂ[1], Liviu CIOVICĂ[2], Marian-Pompiliu CRISTESCU[2], Corina-Ioana CRISTESCU[1]
[1]Academy of Economic Studies, Bucharest
[2]"Lucian Blaga" University of Sibiu
ciovica_laurentiu@yahoo.com, ciovica_liviu@yahoo.com, mp_cristescu@yahoo.com, ci_cristescu@yahoo.com

Human or virtual agents are presented in our lives daily. They serve our purposes and represent us in different many situations. Nowadays the number of virtual agents is increasing daily because they are cheaper, faster and more accurate than human agents. Our aim in this article is to define a new type of intelligent agent called SGIA – Self Growing Intelligent Agent and a new defining language for it. The SGIA agent is an intelligent agent with all the common agents' characteristics and with other special one: that to learn and grow by itself in knowledge and size.

Keywords: *Software Agent, Knowledge Management, Education Process, Language Development*

1 Introduction

Nowadays we need to resolve things faster than ever in different areas of expertise and we need agents to represent us. The solution is to use virtual agents because they are cheap, intelligent, faster, 24x7 present and active.

A lot of research is already done in the field and a lot will be performed because the agents are become a part of our lives, a needful thing without which we cannot perform our daily tasks.

We are introducing a new type of agent called Self Growing Intelligent Agent (SGIA) which besides the common agents' characteristics (autonomy, persistence, and the ability to interact with its environment and others) has other characteristics such as: self-learning and growing mechanisms.

Besides the new introduced agent type we also introduce a new defining language through which the SGIA is to be programmed, defined, trained and through which the agent itself can develop new skills, modules, mechanisms or services in its learning and growing processes.

Studying the way on which the human knowledge is accumulated and grows in size we start defining a new agent type which uses similar techniques and mechanisms for learning and growing.

The article is structured as follows: the second section presents a quick review of intelligent agents, followed by the third section in which we present the SGIA concepts and its characteristics, followed by the 4th section where we present the defining language – the common growing language.

2 Intelligent agents

The KidSim Agent (Smith, Cypher and Spohrer 1994) "Let us define an agent as a persistent software entity dedicated to a specific purpose. 'Persistent' distinguishes agents from subroutines; agents have their own ideas about how to accomplish tasks, their own agendas. 'Special purpose' distinguishes them from entire multifunction applications; agents are typically much smaller."

Autonomous agents are computational systems that inhabit a certain complex and dynamic environment, perceive and act autonomously in this environment and thus achieve its set of goals or tasks for which they were designed [1].

There are three key characteristics of an agent: autonomy, persistence, and the ability to interact with its environment. Autonomy is the agent capability in taking its own decision based on its own experience, environment and also in controlling its own

internal behavior and state. An agent it is also persistent over time and runs continuously within its environment. It has also the capability to interact with the environment – physical or not – through sensors and perform actions through actuators and effectors.
To a greater resemblance to humans an agent must also be capable to reacts at environment changes, to be goal directed, to be adaptable and to be social [2].

3 SGIA – Self Growing Intelligent Agent

Definition 1 The SGIA is an intelligent agent capable, independently, to learn, to assimilate to adapt to its environment and to grow from a small, simple entity – having only sensors, techniques and mechanisms to evolve – to a complex virtual entity.
Definition 2 A SGIA is defined as a virtual entity which with its own technical resources acquires new skills and knowledge throughout his life.
Definition 3 The SGIA growth is the mechanism by which the agent assimilates knowledge and develop new skills, capabilities, services and mechanisms of interaction with the environment independently, influenced directly only by the received education.
Definition 4 The SGIA education consist in adapting it to the virtual environment in which is to operate, however, in view of its nature and its possibility of assimilation. This means to use means, methods and procedures appropriated to the target environment.

Characteristics

According to the above definitions the agent should have, besides his fundamental characteristics, also the next ones:
- Capability to learn by observing, learning and experiencing;
- Capability to adapt to different environments;
- Autonomy and persistence over time and space;
- Sociability – will be able to interact with other agents and communicate with them even if they spoke a different "language";
- Capability to literally grow in size and knowledge, meaning: capability to develop new skills and ways to lean and interact with different agents and environment.

These characteristics are embedded in the agent through its nature and are defined using some of the well-known algorithms. We are not yet decided which the better ones are for accomplish them.

Knowledge representation

Knowledge representation is one of the fundamental problems of artificial intelligence. There are several types of knowledge to be represented: simple facts, complex relationships, natural language syntax rules, associations between related concepts, inheritance hierarchies between classes of objects. Besides the quality of being easy to use, a knowledge base must be easily modified and extended, using a graphical tool, or by automated techniques.
In general, the methods of knowledge representation are based on the use of slots or attribute that store information about certain entities, and links or references to other entities.
A much used method is to represent declarative knowledge representation. In its context the user declares the ontology and specifies instances of data, which is pure knowledge.
The SGIA uses a database system as a representation and reasoning system for its knowledge and besides using attributes, classes and relations between entities (an OOP approach) uses the common growing language for saving the information regarding how the knowledge was obtain and where the knowledge resides. The CGL scripts definitions are kept in files and stored in the agent's memory.
For example in addition to information required for creating a connection to an Oracle database, the agent stores also the steps needed to complete it. In this way the agent knows the steps needed to connect to any database regardless of its nature.
SGIA by its nature, store any information considered useful. Also it saves some partial

information about actions and/or founded models because it can either return to them in the future either trying to obtain missing information from its neighbors.

A piece of information is considered to be useful if the SGIA detects a relevance to its nature or purpose.

For example if the purpose of the agent is to connect to different types of databases for information retrieval and if it finds a piece of information regarding connecting to a HSQL database and it has no knowledge about it, the SGIA detects that has a highly relevance to its nature, collects that information and stores it for later use.

The learning process

In the following lines we describe the learning processes through which the SGIA is acquiring information and builds knowledge:

1. Forms recognition - by computer vision, either by describing the shape, the agent is able to detect - on the basis of experience and previous knowledge base – forms available in the environment and save them.

Even if the form is not completely identified the agent saves all the acquired data and the necessary steps in obtaining that information for its future recognition. Thus, as the agent accumulates more knowledge it can come back to try to recognize it.

2. Experimentation - whether it is produced by the agent in question or by others, experimentation is the process by which an agent tries to make actions based on partial previously accumulated knowledge and based on some statistical analysis it may determine the successful or failure status of the experiment.

3. Cloning and reproduction – is the process by which the agent copies and reproduces, from other agents either knowledge or the necessary steps to discover new knowledge.

4. Achievement of association - in the SGIA context, such processes can be described by: making all connections between existing or new concepts taking into account its own gained experience or the expertise of the other agents.

5. Achievement and storage of information - this is the part of extraction process / generation / acquisition of ontology's agent environment. This procedure can be found in the literature under the name of information extraction or natural language processing

All the procedures explained above, follows two approaches which are generally recognized: symbolic and numeric learning (Adrian A. Hopgood). The symbolic learning is the one through which the systems creates and modifies facts, relations and rules through words and symbols – they can create and modify their one knowledge. The numeric learning affects those systems which use numerical approaches and different types of optimization techniques – this type of learning is using neural networks and genetic algorithms for optimization.

The education process

As we mention in the above sections, the SGIA is desired to become a more human like agent having almost the same techniques in learning and growing – changed to feed its needs as a virtual entity.

Knowing the fact that a human has a teacher from early days also the SGIA can have a teacher, the one which will define the purpose, the skills and the mechanisms through which the agent connects to its future environment.

Through this education process the agent can be trained in discovery and creating its basic skills and its security rules – meanly its borders as what it can do, what can it learn and how much can grow / expand.

This education can form the agent in a good or bad way but all the agent actions are to be restricted either by other agents or by the environment itself.

The growing process

If until now, the agents were growing only in their level and knowledge the SGIA designed to increase both in knowledge and the memory occupied space and also in the number of skills acquired or created.

For example, an agent is first designed to read only online news based on some well-defined criteria either by the user or by its creator. Meanwhile the users want to read

also blog posts. For this new information, the creator should update the agent by adding new functionalities to allow reading blog posts.
If an SGIA will be used, this will be initial designed to read newspapers and educated on how to read any online text / RSS feeds, etc. Thus, if the agent has to read blog posts it can adapt and based on its training can fulfill this task successfully.
To support the need of creating and educating this new tip of agent we had designed a new language, a scripting one, with the help of which the creators can design, define and educate their agents and the agents can grow by defining their new accumulated knowledge, skills, techniques and mechanisms by themselves.
With the help of this language the creator can:

- define the agent and its purpose
- to endow the agent with different mechanisms and initial knowledge
- to define rules constraining the development agent
- to define the security rules of the agent
- to educate the agent regarding to its target environment

4 The Common Growing Language

The Common Growing Language, also called CGL, it's a scripting new language special build in order to define the new introduced agent type. Being only in the pre-alpha we describe only the important, not finished defined, parts.
The language was created using the XText language development framework because we have created also a sophisticated Eclipse-based development environment.
"xText is part of the openArchitectureWare [oAW] project (which is in turn part of Eclipse GMT). oAW provides a set of tools to develop MDSD infrastructures; it helps with meta modeling, constraint checking, code generation and model transformation. More recently a framework has been developed that supports the creation of textual domain-specific languages (DSL): xText" [9]

The new development environment offers the possibility to write agents, compile and build them. The building result is a standard JADE compliant agent having all the options, mechanism that were presented in the previously sections of this article.
Through this new language we want to reduce the amount of work for defining agents and their behaviors through the standard JADE way.
In the next lines we want to present some parts of the defined language:

```
SGIAgent: 'SGIAAgent' name = ID '{'
	internalId = ID
	version = INT
	prettyName = STRING
	internalName = STRING
	birthDate = DATE
	age = INT
	state = AgentState
	'memory' ':' memory = SGIAMemory
	(sensors += Sensor)*
'}' ;
```

An agent has a pretty and internal name with which the agent it is identified by the humans and internally by other agents or by the environment. Also an agent has set of sensors and a memory of its own.

```
SGIAMemory:  '{'
	types += MemoryType
	(properties += Property)*
	'knowledgeBase'  ':'  knowledge  =
KnowledgeBase
'}' ;
```

The SGIA memory is defined by a type (Memory, Cloud, Database or File persistent), some specific properties and a knowledge base.

```
KnowledgeBase: '{'
	'knowledge' ':' '{' (knowledge +=
Knowledge)+ '}'
	created = DATE
	lastUpdate = DATE
		'}' ;
```

The knowledge base is defined as a set of knowledge and some temporal properties.

```
Knowledge: name = ID '{'
	information = ANY_TYPE
	(acquisitions += Acquisition)*
'}' ;
```

The knowledge is defined by information and its acquisition rules.

```
Acquisition : 'acquisition' name = ID
'{'
        'steps'
           (steps += Step)+
        'end'
        ('resetSteps'
           (resetSteps += [Step])+
        'end')?
        'commands'
           (commands += Command)+
        'end'
        (states += State)+
 '}' ;
```

The acquisition is a set of steps, commands and states that can be used to define the way or the process through which an information or an action can be acquired or described.
Being only at pre-alpha release, the language contains only the basic and general structures. The author's intention is to define a full and mature language to be used in production and in many area of expertise.
An example of a SGIA definition. The agent is specialized and trained in reading blog posts:

```
SGIAAgent blogReaderAgent {
       SGIA-0001
1              // the agent version. The
information os hold for
            // updating purposes
       BlogReaderAgent     // the pretty
name. The name seen by the end-user
       SGIA-BRA     //   internal   agent
name
10-10-2010   // the agent birth day. The
date  when  the  agent  was    first  //
activated and registered to agency
       10           // the agent age in
days
       RUNNING      // the agent state
       memory : {
          DATABASE  // memory type
          //     database     connection
properties
          property: {user, root}
          property: {password, secret}
          property: {hostname, localhost}
          property: {driver, jdbcDriver}
          property:   {database,   'blog-
reader-agent-db'}
          // defining knowledge base
          knowledgeBase : {
       knowledge : {
          information   :  read_blog_feed
       // the knowledge information
          acquisition  :  access_blog  {
// the information acquisition
// process
            steps  //  steps  used  in
gathering information

      read_target_blog_url_from_db RTBU
                  access_blog_by_url
ABBU
                  read_blog_feed RBFE
            end
            commands    //    internal
commands used for distinguishing the
                     // global commands
performed within steps.
                  READ_BLOG_URL RBU
                  ACCESS_BLOG   ACB
                  FIND_FEED     FFE
                  READ_FEED     RFE
                  PARSE_FEED    PFE
            end
            //   acquisition   process
states
             state idle
               actions {READ_BLOG_URL}

read_target_blog_url_from_db => active
            end
             state active
               actions {ACCESS_BLOG}
               access_blog_by_url     =>
ready
            end
            state ready
               actions       {FIND_FEED,
READ_FEED, PARSE_FEED}
               read_blog_feed => idle
            end
   }

      }
      10-10-2010   //          knowledge
creation date
      10-10-2010   //   knowledge   last
update date
   }
      }
      '}' ;
```

The example presents brief descriptions of a SGIA. As mentioned already in the paper, the agent stores the acquisitions steps on how a piece of information is retrieved and used. Being suggestively named, the steps, states and actions can be identified later in a similar case of information retrieval.
For example if the SGIA have to access a web page through its URL address it first performs an in-memory search for keywords like: access, read, web, URL, address, etc and it finds the command READ_BLOG_URL. Finding this command, with a high relevance, it takes that piece of

information and the algorithms related to it and try to execute the same steps and algorithms for accessing a web page through its URL. If fails then it tries the next command and so on. The found commands/steps and actions are taken in descending order of their relevance.

5 Conclusion

Today things must be resolved faster than ever. Each one of us must have at least one representative in many areas of expertise. This representative must be cheap, intelligent highly adaptable to continuous changing of its environment and must be persistent in time. The solution is to use an intelligent agent capable to learn and adapt to its environment, easy to program and control.

The SGIA – Self Growing Intelligent Agent is defined to be a virtual intelligent entity capable in self-developing in an independent mode, and to grow in size, knowledge and skills directly influenced by its environment and by its received education and training.

For defining this new agent type a new language was designed and defined. The Common Growing Language (CGL) is a scripting language through which the SGIA can be defined, developed, educated and trained.

The language is intend to be used first by programmers, second by educators / trainers and third by the agent itself in its growing and learning processes.

This new type of agent and its new defined development language wants to open and describe new directions on intelligent agent's research.

References

[1] P. Maes, "Artificial Life Meets Entertainment: Life like Autonomous Agents," *Communications of the ACM*, 1995.

[2] L. Ciovică, L. Ciovică, "SGIA – Self Growing Intelligent Agent," *First International Conference on Modeling and Development of Intelligent Systems*, October 22-25, 2009.

[3] J.M. Alliot, T. Schiex, *Intelligence Artificielle et Informatique Théorique*, Ed. Cepadeus, 1993, First Edition.

[4] M. Moise, *Inteligenţa artificială şi Sistemele Expert*, Ed. Universul Juridic, Bucureşti, 2006.

[5] W. Van de Velde, "Cognitive arhitectures - from knowledge level to structural coupling", In Steels L., *The biology and tehnology of Intelligent Autonomous Agents.*

[6] T. Finin, Y. Labrou, "KQML as an Agent Communication Language", In *Software Agents*. Bradshaw, J. M., MIT Press, Cambridge, MA, pp. 291-316, 1997.

[7] S. Russell, "Rationality and intelligence" *Artificial Intelligence Journal*, Vol. 94, No. 1-2, pp. 57-77, 1997.

[8] M. Wooldridge, N.R. Jennings, "Intelligent Agents: Theory and Practice", *The Knowledge Engineering Review*, Vol. 10, No. 2, pp. 115-152.

[9] oAW xText: A framework for textual DSLs, *Version 1.1, September 22, 2006*, Sven Efftinge, Markus Völter

Vasile-Laurenţiu CIOVICĂ has graduated the Faculty of Science, in 2008 gaining a Bachelor of Science degree in Information Technology with a thesis on Translators and Interpreters for Code Generation and Software Optimization. In 2010 he gained a Master of Management degree in the field of Cybernetics, Statistics and Economic Informatics with a thesis on Intelligent Agents. He is currently a PhD student at Academy of Economic Studies in Bucharest.

Ion – Liviu CIOVICĂ has graduated the Faculty of Science, in 2009 gaining a Bachelor of Science degree in Information Technology with a thesis on "Software application for medical diagnosis assisted by computer". He is currently a student at the Faculty of Science master program, field of Cybernetics, Statistics and Economic Informatics.

Marian-Pompiliu CRISTESCU has graduated the Faculty of Planification and Economic Cybernetics in 1985, he holds a PhD diploma in Economics from 2003, obtained at the Faculty of Cybernetics, Statistic and Economic Informatics from the Academy of Economic Studies in Bucharest. Presently, he is associate professor in Economic Informatics at the Faculty of Economic Sciences - "Lucian Blaga" University of Sibiu.

Corina-Ioana CRISTESCU has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2009, she holds a bachelor degree in Economics, in the Economic Informatics specialty, with a thesis on Distributed Applications Optimization. She is currently a student at the Scientific Economic Informatics Master program.